



ATID Application Development Framework Reference Manual – RFID900Mhz

Revision: Ver. 0.6

Date: January, 2014

ATID Co., Ltd

Table of Contents

Table of Contents	2
Acronym	7
Revision History	8
1 .NET API Reference	9
1.1 Enumerations	10
1.1.1 RFID_RESULT	10
1.1.2 MEM_BANK	11
1.1.3 RFID_READ_TYPE	11
1.1.4 PERMALOCK_FIELD	11
1.1.5 RFID_CALLBACK_TYPE	12
1.1.6 INVENTORIED_STATE	12
1.1.7 SESSION_TYPE	12
1.1.8 SELECT_ACTION	13
1.1.9 SELECT_TARGET	13
1.2 Constants	15
1.2.1 MAX_MASKS_BYTES	15
1.2.2 MODULE_TYPE_NONE	15
1.2.3 MODULE_TYPE_H1000	15
1.2.4 MODULE_TYPE_I2000	15
1.2.5 MODULE_TYPE_I900	15
1.3 Reply Words List	16
1.4 Structures	18
1.4.1 LOCKUNLOCKFIELD	18
1.4.2 RFIDMASKPARAMS	18
1.4.3 RFIDMASKPARAMS_EX	19
1.4.4 RFIDSELMASKPARAMS_EX	20
1.4.5 RFIDCALLBACKDATA	20
1.4.6 AIRDURAPARAMS	21
1.5 Delegates	22
1.5.1 RfidCallbackProc	22
1.6 Constructor	23
1.7 Methods	23
1.7.1 PowerOn	23
1.7.2 PowerOff	23
1.7.3 Open	24
1.7.4 Close	24
1.7.5 IsOpened	24

1.7.6	SetCallback.....	24
1.7.7	GetResult.....	25
1.7.8	ReadEpc.....	26
1.7.9	ReadEpcEx	26
1.7.10	Stop	27
1.7.11	ReadMemBank	28
1.7.12	ReadMemBankEx.....	29
1.7.13	WriteMemBank	30
1.7.14	WriteMemBankEx.....	32
1.7.15	LockField	33
1.7.16	LockFieldEx.....	34
1.7.17	UnLockField	35
1.7.18	UnLockFieldEx.....	36
1.7.19	Permalock	37
1.7.20	PermalockEx.....	38
1.7.21	KillTag	40
1.7.22	KillTagEx.....	40
1.7.23	SetDefault	42
1.7.24	EnableExtendedInformation	42
1.7.25	IsRunning.....	42
1.7.26	GetR1000MacReg.....	43
1.7.27	SetR1000MacReg	43
1.7.28	SetAirDura	44
1.7.29	GetEuroMode.....	44
1.7.30	SetEuroMode	44
1.8	Properties	46
1.8.1	FirmwareVersion	46
1.8.2	HoppingMode	46
1.8.3	InventoryTarget	46
1.8.4	LBTChannelState	47
1.8.5	ChannelState	47
1.8.6	LBTTime	48
1.8.7	OperationTime	48
1.8.8	PowerLevel	48
1.8.9	ProtocolVersion	48
1.8.10	QValue.....	49
1.8.11	Session	49
1.8.12	Selects	49
1.8.13	AirDuty.....	50

1.8.14	ModuleType.....	50
1.8.15	MaxPwr	50
1.8.16	EnableTagFocus.....	50
2	C/C++ API Reference	52
2.1	Enumerations.....	53
2.1.1	RFID_RESULT.....	53
2.1.2	RFIDMEM_BANK.....	54
2.1.3	RFIDREAD_TYPE	54
2.1.4	RFIDPERMALOCK_FIELD	54
2.1.5	RFIDCALLBACKTYPE	55
2.1.6	RFIDINVENTORIEDSTATE	55
2.1.7	RFIDSESSIONTYPE	55
2.2	Constants.....	56
2.2.1	WM_RFID_RESPONSE.....	56
2.2.2	MAX_MASKS_BYTES	56
2.2.3	MODULE_TYPE_NONE	56
2.2.4	MODULE_TYPE_H1000.....	56
2.2.5	MODULE_TYPE_I2000	56
2.2.6	MODULE_TYPE_I900.....	56
2.3	Reply Words List	57
2.4	Structures.....	59
2.4.1	RFIDLOCK_UNLOCK_FIELD	59
2.4.2	RFIDMASK_PARAMS	59
2.4.3	RFIDMASK_PARAMS_EX	60
2.4.4	RFID_MASK_PARAMS_EX2.....	61
2.4.5	RFIDCALLBACKDATA.....	62
2.4.6	RFIDAIRDURAPARAMS.....	62
2.5	Callback function definition.....	63
2.5.1	RFIDCALLBACK	63
2.6	Methods.....	64
2.6.1	RfidPowerOn	64
2.6.2	RfidPowerOff	64
2.6.3	RfidOpen	64
2.6.4	RfidClose	65
2.6.5	RfidIsOpened	65
2.6.6	RfidSetCallback.....	65
2.6.7	RfidSetHwnd.....	66
2.6.8	RfidGetResult.....	66
2.6.9	RfidReadEpc.....	67

2.6.10	RfidReadEpcEx	67
2.6.11	RfidReadEpcEx2.....	68
2.6.12	RfidStop	69
2.6.13	RfidStopEx	69
2.6.14	RfidReadMemBank	69
2.6.15	RfidReadMemBankEx.....	70
2.6.16	RfidReadMemBankEx2	71
2.6.17	RfidWriteMemBank	72
2.6.18	RfidWriteMemBankEx.....	73
2.6.19	RfidWriteMemBankEx2	74
2.6.20	RfidLockField	75
2.6.21	RfidLockFieldEx	76
2.6.22	RfidLockFieldEx2.....	76
2.6.23	RfidUnLockField	77
2.6.24	RfidUnLockFieldEx.....	78
2.6.25	RfidUnLockFieldEx2.....	79
2.6.26	RfidPermalock.....	79
2.6.27	RfidPermalockEx	80
2.6.28	RfidPermalockEx2.....	81
2.6.29	RfidKillTag	82
2.6.30	RfidKillTagEx.....	83
2.6.31	RfidKillTagEx2	83
2.6.32	RfidSetDefault.....	84
2.6.33	RfidSetExtInfoEnable.....	85
2.6.34	RfidGetQValue	85
2.6.35	RfidSetQValue.....	85
2.6.36	RfidGetOperationTime.....	86
2.6.37	RfidSetOperationTime.....	86
2.6.38	RfidGetPowerLevel.....	86
2.6.39	RfidSetPowerLevel.....	87
2.6.40	RfidGetSession.....	87
2.6.41	RfidSetSession	87
2.6.42	RfidGetInventoryTarget.....	88
2.6.43	RfidSetInventoryTarget	88
2.6.44	RfidGetLbtChState	89
2.6.45	RfidSetLbtChState	90
2.6.46	RfidGetChannelState.....	90
2.6.47	RfidSetChannelState	90
2.6.48	RfidGetLbtTime	91

2.6.49	RfidSetLbtTime	91
2.6.50	RfidSetAirDuty	92
2.6.51	RfidGetFirmwareVersion	92
2.6.52	RfidGetModuleType	93
2.6.53	RfidGetXcvrMaxPwr	93
2.6.54	RfidIsRunning	93
2.6.55	RfidGetR1000MacReg	94
2.6.56	RfidSetR1000MacReg	94
2.6.57	RfidSetTagFocus	94
2.6.58	RfidGetTagFocusStatus	95
2.6.59	RfidSetAirDura	95
2.6.60	RfidGetEuroMode	96
2.6.61	RfidSetEuroMode	96

Acronym

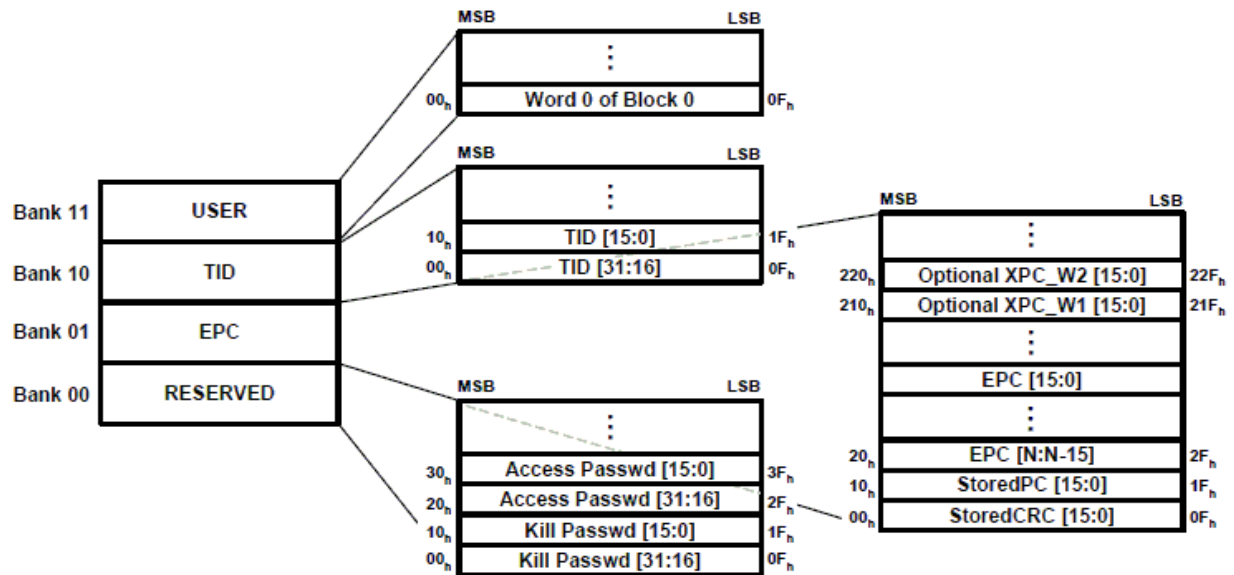
modules	descriptions
AADF	ATID Application Development Framework
EPC	Electronic Product Code

Revision History

Version	Date	Reason	Description	Author
0.1	2012/01/17	Draft		Y. J. CHO
0.2	2013/01/27	Update	<ul style="list-style-type: none"> - IsRunning 추가. - Stop(bool block) 추가. - GetModuleType, GetMaxPwr 추가. - LBT Channel state 설명 보강. 	Y. J. CHO
0.3	2013/02/26	Update	<ul style="list-style-type: none"> - R1000 레지스터 컨트롤함수 추가. 	Y. J. CHO
0.4	2013/05/28	Update	<ul style="list-style-type: none"> - SetAirDura, TagFocus 함수 추가. - .NET API 클래스 생성자 추가. - 오타 수정 	Y. J. CHO
0.5	2013/09/12	Update	<ul style="list-style-type: none"> - PowerLevel의 Max를 30으로 변경 - RFIDMASKPARAMS_EX의 MaskPattern 멤버 초기화 방법 설명 추가. - AIRDURAPARAMS 구조체의 멤버의 min, max 설명 추가. 	Y. J. CHO
0.6	2014/01/10	Update	<ul style="list-style-type: none"> - LBTChannelState 함수와 같은 기능을 하는 ChannelState 함수 추가. - Europe향 모드 변경 함수 추가. 	Y. J. CHO

1 .NET API Reference

- ISO 18000 6C(EPC Class 1 Generation 2) Logical Memory Map



1.1 Enumerations

1.1.1 RFID_RESULT

함수의 호출 결과를 나타낸다.

- **RFID_RESULT_SUCCESS**

기능 수행 성공.

- **RFID_RESULT_OUTOFMEMORY**

자원 할당 실패.

- **RFID_RESULT_INVALID_ARGS**

유효하지 않은 parameter.

- **RFID_RESULT_FAILURE**

기능 수행 실패.

- **RFID_RESULT_UNEXPECTED**

알 수 없는 Error 발생.

- **RFID_RESULT_ALREADY_OPENED**

UHF RFID 장치 port가 이미 열려 있음.

- **RFID_RESULT_INVALID_DEVICE**

지원하지 않는 장치 타입.

- **RFID_RESULT_NOT_CONNECTED**

장치 open 후 정상적으로 반응 하지 않음.

- **RFID_RESULT_NOT_DETECT**

Tag를 찾지 못하고 명령이 종료됨.

- **RFID_RESULT_ACCESS_ERROR**

잘못된 패스워드로 접근 시도 또는 존재 하지 않는 영역에 접근 시도.

- **RFID_RESULT_NOT_OPENED**

Open을 호출하지 않고 기능 수행을 시도 함.

- **RFID_RESULT_COMMAND_ERROR**

명령 실행 중 에러 발생, 또는 명령 실행이 종료 되지 않은 상태에서 다른 명령을 받음.

- **RFID_RESULT_LOW_BATTERY**

모듈에 공급되는 전원의 전압이 낮아 동작 할 수 없음.

- **RFID_RESULT_UNKNOWN**

알 수 없는 Error 발생.

- **RFID_RESULT_NOT_SUPPORTED**

지원 되지 않는 명령.

- **RFID_RESULT_STOPPED**

명령 실행이 중단 됨.

- **RFID_RESULT_POWER_OFF**

UHF RFID 장치의 전원이 off되어 있음.

1.1.2 MEM_BANK

RFID 장치가 access할 Tag의 Memory bank

- **RESERVED**

kill password 또는 access password가 저장되는 memory.

- **EPC**

StoredCRC, EPC, XPC 정보들이 저장되는 memory.

- **TID**

ISO/IEC 15963 allocation class identifier, Tag 일련 번호 등이 저장된다.

- **USER**

user specific data storage.

1.1.3 RFID_READ_TYPE

Tag reading(Inventory) 시 적용될 single, multiple mode.

- **EPC_GEN2_MULTI_TAG**

다수의 Tag를 동시에 읽는다.

- **EPC_GEN2_ONE_TAG**

한 장의 Tag만 읽고 읽기를 중단 한다.

1.1.4 PERMALOCK_FIELD

Permalock을 적용 시킬 field

- **ACCESS_PASSWORD**

Tag의 Access password field

- **KILL_PASSWORD**

Tag의 Kill password field

- **EPC**

Tag의 EPC bank

- **TID**

Tag의 TID bank

- **USER**

Tag의 USER bank

1.1.5 RFID_CALLBACK_TYPE

Callback delegate가 실행되는 이유를 나타내는 값

- RFIDCALLBACKTYPE_DATA

Tag로부터 Memory data를 읽었기 때문에 발생함.

- RFIDCALLBACKTYPE_REPLY

RFID 장치로부터 명령의 실행 결과를 받았을 때 발생함.

1.1.6 INVENTORIED_STATE

ReadEpc(Inventory)를 수행할 때 Target으로 사용할 Tag의 Inventoried state.

- STATE_A

Inventoried state가 A 인 Tag만 Inventory 함.

- STATE_B

Inventoried state가 B 인 Tag만 Inventory 함.

- STATE_AB

Inventoried state가 A또는B 인 Tag를 Inventory 함.

1.1.7 SESSION_TYPE

ReadEpc(Inventory)를 수행할 때 사용할 Session

- SESSION_0

S0

- SESSION_1

S1

- SESSION_2

S2

- SESSION_3

S3

1.1.8 SELECT_ACTION

xxxEx() 함수들을 사용해서 적용할 SELECT 명령.

- **ACTION_0**

000

- **ACTION_1**

001

- **ACTION_2**

010

- **ACTION_3**

011

- **ACTION_4**

100

- **ACTION_5**

101

- **ACTION_6**

110

- **ACTION_7**

111

1.1.9 SELECT_TARGET

xxxEx() 함수들을 사용해서 SELECT 명령을 적용할 태그의 SL flag 또는 inventoried flag.

- **S0**

S0

- **S1**

S1

- **S2**

S2

- **S3**

S3

- **SL**

SL

SELECT_ACTION, SELECT_TARGET의 조합으로 아래 표와 같이 Tag의 상태를 변경 할 수 있다.

	If SELECT_TARGET is SL		If SELECT_TARGET are between S0 ~ S3	
	Matching	Non-matching	Matching	Non-matching
ACTION_0	Assert SL	De-assert SL	→ A	→ B
ACTION_1	Assert SL	No Action	→ A	No Action
ACTION_2	No Action	De-assert SL	No Action	→ B
ACTION_3	Negate SL	No Action	Negate	No Action
ACTION_4	De-assert SL	Assert SL	→ B	→ A
ACTION_5	De-assert SL	No Action	→ B	No Action
ACTION_6	No Action	Assert SL	No Action	→ A
ACTION_7	No Action	Negate SL	No Action	Negate

1.2 Constants

1.2.1 MAX_MASKS_BYTES

Masking pattern으로 사용할 수 있는 최대 바이트 수

- **Public const int** **MAX_MASKS_BYTES** **32**

1.2.2 MODULE_TYPE_NONE

RFID 모듈이 장착되어 있지 않음.

- **Public const int** **MODULE_TYPE_H1000** **0x30**

1.2.3 MODULE_TYPE_H1000

2013년 1월 현재, 일본을 제외한 국가에 해당하는 RFID 모듈

Public const int **MODULE_TYPE_H1000** **0x31**

1.2.4 MODULE_TYPE_I2000

2013년 1월 현재, 일본에만 해당하는 RFID 모듈

일본의 변경된 전파법이 적용됨.

Public const int **MODULE_TYPE_I2000** **0x32**

1.2.5 MODULE_TYPE_I900

2013년 1월 현재, AT980에만 장착되는 RFID 모듈

Public const int **MODULE_TYPE_I900** **0x33**

1.3 Reply Words List

SyncMode를 사용하지 않고 Tag Operation 함수들을 실행하면 Callback delegate를 통해서 아래와 같은 문자열을 받는다.

- **“Other Error”**

알 수 없는 에러 발생.

- **“Memory Overrun”**

존재 하지 않는 Memory 영역에 접근을 시도 함.

- **“Memory Locked”**

Tag가 잠겨 있음.

- **“Insufficient Tag Power”**

Tag의 신호가 약해서 기능 수행 실패.

- **“Non-specific Error”**

알 수 없는 에러 발생.

- **“Check Antenna”**

RFID 장치와 Antenna간의 연결 상태 불량.

- **“Try after cooled”**

RFID 장치가 과열 됨.

- **“Insufficient PDA Power”**

PDA 메인 배터리 전원이 낮아서 기능 수행 실패.

- **“Not Supported”**

지원되지 않는 기능.

- **“Not Connected”**

RFID 장치가 open 이후 정상적으로 반응하지 않음.

- **“Not Opened”**

Open 함수를 수행하지 않고 기능 함수를 호출함.

- **“Bad Access Password”**

잘못된 Access Password로 Tag에 접근 함.

- **“Invalid Parameter”**

유효하지 않은 parameter

- **“Command Error”**

명령 수행 도중 에러 발생 또는 이전 명령이 종료 되기 전에 또 다른 명령을 실행 함.

- **“Success”**

성공적으로 함수 실행 완료.

- **“Not Detect”**

Tag를 찾지 못하고 명령이 종료 됨.

- **“Multi Read Stop”**

ReadEpc(multiple) 명령이 Stop 명령에 의해서 종료 됨.

- **“EAS”**

NXP Tag의 EAS가 감지 됨.

1.4 Structures

1.4.1 LOCKUNLOCKFIELD

Tag의 각 field를 lock 또는 unlock 하기 위한 정보를 저장한다.
True로 설정된 멤버에 대해서만 lock 또는 unlock이 적용된다.

Public struct **LOCKUNLOCKFIELD**

```
{  
    bool bAccessPassword;  
    bool bEPC;  
    bool bKillPassword;  
    bool bTID;  
    bool bUSER;  
};  
  
- bAccessPassword  
    Tag의 access password field  
- bEPC  
    Tag의 EPC bank  
- bKillPassword  
    Tag의 kill password field  
- bTID  
    Tag의 TID bank  
- bUSER  
    Tag의 USER bank.
```

Ex) Access Password와 EPC를 Lock 또는 Unlock하기 위해서는 아래와 같이 구성한다.

```
LOCKUNLOCKFIELD Lock = new LOCKUNLOCKFIELD();
```

```
Lock.bAccessPassword = true;
```

```
Lock. bEPC = true;
```

```
Lock. bKillPassword = false;
```

```
Lock. bTID = false;
```

```
Lock. bUSER = false;
```

1.4.2 RFIDMASKPARAMS

태그에 대한 작업을 수행하는 명령 호출 시에 조건에 해당되는 태그에만 반응 하도록 할 때 사용한다.

Public struct [RFIDMASKPARAMS](#)

```
{
    MEM\_BANK MemBank;
    uint nOffSet;
    String MaskPattern;
};
```

- **MemBank**

Mask로 사용될 태그의 Memory Bank

- **nOffSet**

선택한 Memory Bank에서 MaskPattern의 nibble offset

- **MaskPattern**

4bit로 지정된 masking pattern string

Ex) TID bank가 "E2006004015F325A"로 입력되어 있는 태그를 TID bank를 Mask로 사용하여 읽기 위해서는 아래와 같이 구성할 수 있다..

```
RFIDMASKPARAMS Mask = new RFIDMASKPARAMS();
```

```
Mask.MemBank = MEM_BANK.TID;
```

```
Mask.nOffSet = 1;
```

```
Mask.MaskPattern = "2006";
```

또는

```
RFIDMASKPARAMS Mask = new RFIDMASKPARAMS();
```

```
Mask.MemBank = MEM_BANK.TID;
```

```
Mask.nOffSet = 11;
```

```
Mask.MaskPattern = "F325";
```

1.4.3 **RFIDMASKPARAMS_EX**

Masking pattern을 bit 단위로 적용할 때 사용한다.

Public struct [RFIDMASKPARAMS_EX](#)

```
{
    MEM\_BANK MemBank;
    uint nOffSet;
    uint nBits;
    byte\[\] MaskPattern;
};
```

- **MemBank**

Mask로 사용될 태그의 Memory Bank

- **nOffSet**

선택한 Memory Bank에서 MaskPattern의 bit offset

- **nBits**

masking pattern으로 사용될 MaskPattern의 bit 수.

- **MaskPattern**

masking pattern byte array

MAX_MASK_BYTES 상수를 사용하여 초기화 해야 한다.

1.4.4 RFIDSELMASKPARAMS_EX

Bit 단위의 Masking pattern을 사용하여 SELECT 명령을 적용할 때 사용된다.

Public struct [RFIDSELMASKPARAMS_EX](#)

```
{
    SELECT_ACTION ActionCode;
    uint Bits;
    byte[] MaskPattern;
    MEM_BANK MemBank;
    uint OffSet;
    SELECT_TARGET SelectTarget;
};
```

- **ActionCode**

Tag의 flag에 적용할 Select command

- **Bits**

masking pattern으로 사용될 MaskPattern의 bit 수.

- **MaskPattern**

masking pattern byte array

MAX_MASK_BYTES 상수를 사용하여 초기화 해야 한다.

- **MemBank**

Mask로 사용될 태그의 Memory Bank

- **OffSet**

선택한 Memory Bank에서 MaskPattern의 bit offset

- **SelectTarget**

Select command가 적용될 Tag의 flag

1.4.5 RFIDCALLBACKDATA

RFID 장치로부터 응답을 받았을 때 delegate를 통해서 받는 구조체.

Delegate가 호출되면 이 구조체의 멤버들을 가지고 GetResult 함수를 사용하여 데이터를 읽는다.

Public struct **RFIDCALLBACKDATA**

```
{
    RFID_CALLBACK_TYPE CallbackType;
    IntPtr IParam
    IntPtr wParam;
};
```

- **CallbackType**

Delegate 실행의 원인.(Data or Reply)

반드시 **GetResult** 함수의 파라미터로 전달해야 한다.

- **IParam**

RFID 장치로부터 데이터를 읽어 오기 위한 첫번째 파라미터

반드시 **GetResult** 함수의 파라미터로 전달해야 한다.

- **wParam**

RFID 장치로부터 데이터를 읽어 오기 위한 두번째 파라미터

반드시 **GetResult** 함수의 파라미터로 전달해야 한다.

1.4.6 AIRDURAPARAMS

모듈의 Tx Cycle 동작 시간을 조정할 때 사용하는 구조체.

Public struct **AIRDURAPARAMS**

```
{
    uint nOffMs;
    uint nOnMs;
};
```

- **nOffMs**

Tx Cycle에서의 쉬는 시간.(millisecond)

Min:0ms ~ Max:4000ms

- **nOnMs**

Tx Cycle에서의 동작 시간(millisecond)

Min:0ms ~ Max:4000ms

1.5 Delegates

1.5.1 RfidCallbackProc

RFID 장치로부터 응답을 받았을 때 실행되는 delegate

Public delegate void **RfidCallbackProc**([RFIDCALLBACKDATA](#) CallbackData);

- CallbackData

RFID 장치로부터 받은 응답이 들어 있는 구조체.

1.6 Constructor

RfidApi 클래스의 새 인스턴스를 초기화 합니다.

```
public RfidApi()
```

1.7 Methods

1.7.1 PowerOn

UHF RFID 장치의 전원을 인가한다.

```
RFID_RESULT PowerOn();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

UHF RFID 장치를 사용하기 위해서는 반드시 호출해야 한다.

1.7.2 PowerOff

UHF RFID 장치에 인가된 전원을 제거한다.

```
RFID_RESULT PowerOff();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

UHF RFID 장치 사용이 끝나면, 이 함수를 호출하여 전원을 해제 한다.

1.7.3 Open

시스템 자원을 할당하고, UHF RFID 장치의 port를 open 한다.

```
RFID_RESULT Open();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

1.7.4 Close

UHF RFID 장치의 port를 close하고, 할당된 자원을 해제한다.

```
RFID_RESULT Close();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

1.7.5 IsOpened

UHF RFID 장치의 port가 open되어 있는지 확인한다.

```
BOOL IsOpened()
```

Parameters

None

Return Values

UHF RFID 장치의 open 상태.

True:Opend, False:Closed

1.7.6 SetCallback

RFID 장치에서 응답을 받았을 때 호출될 callback delegate를 등록한다.

```
RFID_RESULT SetCallback (  
    RfidCallbackProc CallbackProc  
);
```


Parameters

CallbackProc

RfidCallbackProc delegate 함수

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Open() 을 호출한 이후에 호출해야 한다.

1.7.7 GetResult

Tag Data 또는 RFID 장치로부터의 응답을 읽는다.

```
RFID_RESULT GetResult (
    String Result,
    RFID_CALLBACK_TYPE CallbackType,
    IntPtr wParam,
    IntPtr lParam
);
```

Parameters

Result

Tag로부터 읽은 Data 또는 Reply가 저장될 변수.

CallbackType

Callback delegate가 실행될 때 넘겨 받은 RFIDCALLBACKDATA의 CallbackType을 그대로 CallbackType parameter 값으로 설정해준다. 또한 응용프로그램에서는 RFIDCALLBACKDATA의 CallbackType의 내용을 보고 callback delegate의 호출이 data 또는 응답(reply)인지 파악하고 해당 type에 맞게 처리한다.

wParam

Callback delegate가 호출될 때 parameter로 전달 받은 RFIDCALLBACKDATA의 wParam을 그대로 설정해준다.

lParam

Callback delegate가 호출될 때 parameter로 전달 받은 RFIDCALLBACKDATA의 lParam을 그대로 설정해준다.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

이 함수는 반드시 사용자가 등록한 delegate 함수 내에서 호출해야 한다.

1.7.8 ReadEpc

Tag로부터 EPC 데이터를 읽는다. (Inventory)

```
RFID_RESULT ReadEpc(
    bool bSyncMode,
    RFID_READ_TYPE ReadType,
    String EpcData
);
RFID_RESULT ReadEpc(
    bool bSyncMode,
    RFID_READ_TYPE ReadType,
    ref RFIDMASKPARAMS Mask,
    String EpcData
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

ReadType

Singe Tag Reading 또는 Multiple Tag Reading 모드.

Mask

Inventory에 사용될 masking pattern. (nibble단위의 masking pattern)

EpcData

Sync 모드로 호출 했을 때, EPC 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

1.7.9 ReadEpcEx

Tag로부터 EPC 데이터를 읽는다. (Inventory)

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT ReadEpcEx(
    bool bSyncMode,
    RFID_READ_TYPE ReadType,
    String EpcData
);
RFID_RESULT ReadEpcEx(
    bool bSyncMode,
```

```

RFID_READ_TYPE ReadType,
ref RFIDMASKPARAMS_EX Mask,
String EpcData
);
RFID_RESULT ReadEpcEx(
    bool bSyncMode,
    RFID_READ_TYPE ReadType,
    ref RFIDSELMASKPARAMS_EX[] Masks,
    uint NumberOfMasks
    String EpcData
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

ReadType

Singe Tag Reading 또는 Multiple Tag Reading 모드.

Mask

Inventory에 사용될 bit단위의 masking pattern.

Masks

Inventory에 사용될 bit단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

EpcData

Sync 모드로 호출 했을 때, EPC 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Masking pattern의 배열을 사용하는 경우에는 Sync 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.10 Stop

UHF RFID 장치에서 Async 모드로 수행중인 작업을 중단 시킨다.

```

RFID_RESULT Stop();
RFID_RESULT Stop(bool Block);

```

Parameters

Block

동작 모드

True : stop operation이 종료될 때까지 대기하고 리턴된다.(최대 5초)

False : stop 명령을 전송하고 함수가 즉시 리턴된다.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

1.7.11 ReadMemBank

Tag의 특정 memory bank로부터 word(2bytes) 단위로 data를 읽는다.

RFID_RESULT ReadMemBank(

Bool bSyncMode,

MEM_BANK MemBank,

Uint nWordPtr,

Uint nWordCount,

Bool bContinuous,

String AccessPassword,

String MemBankData

);

RFID_RESULT ReadMemBank(

Bool bSyncMode,

MEM_BANK MemBank,

Uint nWordPtr,

Uint nWordCount,

Bool bContinuous,

String AccessPassword,

Ref RFIDMASKPARAMS Mask,

String MemBankData

);

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

읽을 Memory Bank Type

nWordPtr

읽을 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

nWordCount

읽을 data 크기. (Min:0 ~ Max:255)

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

Mask

ReadMemBank에 사용될 nibble 단위의 masking pattern.

MemBankData

Sync 모드로 호출 했을 때, MemBank 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

1.7.12 ReadMemBankEx

Tag의 특정 memory bank로부터 word(2bytes) 단위로 data를 읽는다.

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT ReadMemBankEx(
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    Uint nWordCount,
    Bool bContinuous,
    String AccessPassword,
    String MemBankData
);

RFID_RESULT ReadMemBankEx(
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    Uint nWordCount,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS_EX Mask,
    String MemBankData
);

RFID_RESULT ReadMemBankEx(
    Bool bSyncMode,
    MEM_BANK MemBank,
```

```

    Uint nWordPtr,
    Uint nWordCount,
    Bool bContinuous,
    String AccessPassword,
    RFIDSELMASKPARAMS_EX[] Masks,
    Uint NumberOfMasks,
    String MemBankData
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

읽을 Memory Bank Type

nWordPtr

읽을 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

nWordCount

읽을 data 크기. (Min:0 ~ Max:255)

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

Mask

bit 단위의 masking pattern.

Masks

bit 단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

MemBankData

Sync 모드로 호출 했을 때, MemBank 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.13 WriteMemBank

Tag의 특정 memory bank에 word(2bytes) 단위로 data를 기록한다.

```
RFID_RESULT WriteMemBank(
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    String Data,
    Bool bContinuous,
    String AccessPassword,
);

RFID_RESULT WriteMemBank (
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    String Data,
    Bool bContinuous,
    String AccessPassword,
    Ref RFIDMASKPARAMS Mask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

데이터를 기록할 Memory Bank Type

nWordPtr

데이터를 기록할 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

Data

기록할 데이터.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

Mask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

1.7.14 WriteMemBankEx

Tag의 특정 memory bank에 word(2bytes) 단위로 data를 기록한다.
SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT WriteMemBankEx (
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    String Data,
    Bool bContinuous,
    String AccessPassword,
);

RFID_RESULT WriteMemBankEx (
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    String Data,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS_EX Mask,
);

RFID_RESULT WriteMemBankEx (
    Bool bSyncMode,
    MEM_BANK MemBank,
    Uint nWordPtr,
    String Data,
    Bool bContinuous,
    String AccessPassword,
    RFIDSELMASKPARAMS_EX[] Masks,
    Uint NumberOfMasks
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

데이터를 기록할 Memory Bank Type

nWordPtr

데이터를 기록할 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

Data

기록할 데이터.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

Mask

bit 단위의 masking pattern.

Masks

bit 단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.15 LockField

Tag의 특정 field를 lock 한다. Lock이 된 field 접근시에는 Access password를 필요로 한다.

```
RFID_RESULT LockFiled(
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD LockField,
    Bool bContinuous,
    String AccessPassword,
);

RFID_RESULT LockFiled (
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD LockField,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS Mask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

LockField

Lock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

Mask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.16 LockFieldEx

Tag의 특정 field를 lock 한다. Lock이 된 field 접근시에는 Access password를 필요로 한다.

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT LockFiledEx (
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD LockField,
    Bool bContinuous,
    String AccessPassword,
);

RFID_RESULT LockFiledEx (
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD LockField,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS_EX Mask,
);

RFID_RESULT LockFiledEx (
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD LockField,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDSELMASKPARAMS_EX[] Masks,
```

`uint` NumberOfMasks

);

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

LockField

Lock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

Mask

bit 단위의 masking pattern.

Masks

bit 단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.17 UnLockField

Tag의 특정 field를 unlock 한다.

```
RFID_RESULT UnLockFiled(
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD UnLockField,
    Bool bContinuous,
    String AccessPassword,
);

RFID_RESULT UnLockFiled (
    Bool bSyncMode,
    ref LOCKUNLOCKFIELD UnLockField,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS Mask,
```

);

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

UnLockField

UnLock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

Mask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.18 UnLockFieldEx

Tag의 특정 field를 lock 한다

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

RFID_RESULT UnLockFiledEx (

Bool bSyncMode,

ref LOCKUNLOCKFIELD UnLockField,

Bool bContinuous,

String AccessPassword,

);

RFID_RESULT UnLockFiledEx (

Bool bSyncMode,

ref LOCKUNLOCKFIELD UnLockField,

Bool bContinuous,

String AccessPassword,

ref RFIDMASKPARAMS_EX Mask,

);

RFID_RESULT UnLockFiledEx (

Bool bSyncMode,

ref LOCKUNLOCKFIELD UnLockField,

```

    Bool bContinuous,
    String AccessPassword,
    ref RFIDSELMASKPARAMS_EX[] Masks,
    uint NumberOfMasks
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

LockField

UnLock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

Mask

bit 단위의 masking pattern.

Masks

bit 단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.19 Permalock

Tag의 특정 field를 Permalock 한다.

영구적으로 lock이 된 field는 unlock으로 secured 상태를 해제할 수 없다. Secured 상태는 tag의 password가 enable 되어 있어 tag 접근 시 password를 필요로 하는 상태를 의미한다.

RFID_RESULT Permalock (

Bool bSyncMode,

PERMALOCK_FIELD PermalockField,

Bool bSecured,

Bool bContinuous,

String AccessPassword,

```
);
RFID_RESULT Permalock (
    Bool bSyncMode,
    PERMALOCK_FIELD PermalockField,
    Bool bSecured,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS Mask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

PermalockField

Permalock을 적용할 field

bSecured

Secured 상태.

True로 설정되면 영구히 access password를 가지고 Tag에 access 해야한다.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

Mask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 secured 상태를 영구히 변경할 수 없으므로 주의해야 한다.

1.7.20 PermalockEx

Tag의 특정 field를 permalock 한다

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT PermalockEx (
    Bool bSyncMode,
    ref PERMALOCK_FIELD PermalockField,
    Bool bSecured,
```

```

    Bool bContinuous,
    String AccessPassword,
);
RFID_RESULT PermalockEx (
    Bool bSyncMode,
    ref PERMALOCK_FIELD PermalockField,
    Bool bContinuous,
    String AccessPassword,
    ref RFIDMASKPARAMS_EX Mask,
);
RFID_RESULT PermalockEx (
    Bool bSyncMode,
    ref PERMALOCK_FIELD PermalockField,
    Bool bContinuous,
    String AccessPassword,
    RFIDSELMASKPARAMS_EX[] Masks,
    uint NumberOfMasks
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

LockField

Permalock을 적용할 field

bSecured

Secured 상태.

True로 설정되면 영구히 access password를 가지고 Tag에 access 해야한다.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

Mask

bit 단위의 masking pattern.

Masks

bit 단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.21 KillTag

Tag를 UHF RFID 장치에 반응하지 않도록 Kill한다. Kill password가 "00000000"인 경우에는 kill을 할 수 없다. Kill 명령을 수행하기 위해 tag는 "00000000"가 아닌 다른 password가 설정이 되어 있어야 한다. 한 번 killed 된 tag는 다시 복구할 수 없다.

```
RFID_RESULT KillTag (
    Bool bSyncMode,
    String KillPassword,
    Bool bContinuous,
);

RFID_RESULT KillTag (
    Bool bSyncMode,
    String KillPassword,
    Bool bContinuous,
    ref RFIDMASKPARAMS Mask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

KillPassword

Tag에 저장되어 있는 Kill Password

bContinuous

Continuous 모드. (True=Enable, False=Disable)

Mask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 다시 복구 할 수 없으므로 주의해야 한다.

1.7.22 KillTagEx

Tag를 UHF RFID 장치에 반응하지 않도록 Kill한다. Kill password가 "00000000"인 경우에는 kill을 할 수 없다. Kill 명령을 수행하기 위해 tag는 "00000000"가 아닌 다른 password가 설정이 되어 있어야 한다. 한 번 killed 된 tag는 다시 복구할 수 없다.

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT KillTagEx (
    Bool bSyncMode,
    String KillPassword,
    Bool bContinuous,
);

RFID_RESULT KillTagEx (
    Bool bSyncMode,
    String KillPassword,
    Bool bContinuous,
    ref RFIDMASKPARAMS_EX Mask,
);

RFID_RESULT KillTagEx (
    Bool bSyncMode,
    String KillPassword,
    Bool bContinuous,
    RFIDSELMASKPARAMS_EX[] Masks,
    uint NumberOfMasks
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

KillPassword

Tag에 저장되어 있는 Kill Password

bContinuous

Continuous 모드. (True=Enable, False=Disable)

Mask

bit 단위의 masking pattern.

Masks

bit 단위의 masking pattern의 배열.

NumberOfMasks

Masks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 다시 복구 할 수 없으므로 주의해야 한다.

3번째 오버로드 함수는 모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.23 SetDefault

UHF RFID 장치의 설정 값들을 초기값으로 되돌린다.

```
RFID_RESULT SetDefault ();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

1.7.24 EnableExtendedInformation

Tag를 읽을 때 RSSI 값을 부가적으로 읽도록 설정하거나 해제한다.

```
RFID_RESULT EnableExtendedInformation (  
    Bool bEnable  
);
```

Parameters

bEnable

Extended information(RSSI) 설정 여부. (True:On, False:Off)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

1.7.25 IsRunning

Rfid 모듈이 Operation 진행 중인지 여부를 반환한다.

```
bool IsRunning ( );
```

Parameters

None

Return Values

모듈이 operation을 진행 중이라면 true를 그렇지 않다면 false를 반환한다.

1.7.26 GetR1000MacReg

R1000의 레지스터 값을 읽는다.

```
bool GetR1000MacReg (  
    ushort Reg,  
    ref uint Value  
);
```

Parameters

Reg

레지스터 주소(별도의 문서 참조)

Value

레지스터 값이 저장될 변수

Return Values

성공적으로 수행되면 true를 리턴한다.

Notes

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.27 SetR1000MacReg

R1000의 레지스터에 값을 기록한다.

```
bool SetR1000MacReg (  
    ushort Reg,  
    uint Value  
);
```

Parameters

Reg

레지스터 주소(별도의 문서 참조)

Value

레지스터에 저장할 값이 저장된 변수

Return Values

성공적으로 수행되면 true를 리턴한다.

Notes

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.7.28 SetAirDura

모듈의 Tx Cycle 동작 시간을 조정한다.

```
RFID_RESULT SetAirDura (
    AIRDURAPARAMS Value
);
```

Parameters

Value

Tx Cycle의 on시간과 Off시간이 저장되어 있는 구조체 변수

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴한다.

Notes

MODULE_TYPE_I2000일 경우, Tx Cycle 동작시간의 기본값은 On(200ms), Off(0ms)이다.

1.7.29 GetEuroMode

모듈 타입이 MODULE_TYPE_H1000이고 Europe형일 경우에 모듈의 동작 방식을 읽어 온다.

```
bool GetEuroMode (
    ref uint mode
);
```

Parameters

mode

1 : 4개 채널 hopping

2 : 4개 채널 중 1개 채널이 랜덤으로 선택되어 고정.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴 한다.

Notes

1.7.30 SetEuroMode

모듈 타입이 MODULE_TYPE_H1000이고 Europe형일 경우에 모듈의 동작 방식을 설정한다.

```
bool SetEuroMode (
    uint mode
);
```

Parameters

mode

1 : 4개 채널 hopping

2 : 4개 채널 중 1개 채널이 랜덤으로 선택되어 고정.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴 한다.

Notes

1.8 Properties

1.8.1 FirmwareVersion

UHF RFID 장치의 Firmware Version을 읽는다.

```
string DeviceID{ get; }
```

Value

Firmware Version

1.8.2 HoppingMode

UHF RFID 장치의 Hopping Mode를 읽는다.

```
HOPPING_MODE HoppingMode{ get; }
```

Value

Hopping Mode

1.8.3 InventoryTarget

ReadEpc 수행 시, 사용할 Tag의 Inventoried State를 설정하거나 읽는다.

```
INVENTORIED_STATE InventoryTarget{ get; set;}
```

Value

Inventory Target

1.8.4 LBTChannelState

LBT Channel을 사용자가 구성하거나 읽는다. Japan, Europe type에 해당됨.
모듈별 사용 주파수와 채널 수는 아래의 표와 같다.

MODULE_TYPE_H1000		MODULE_TYPE_I2000	
Europe	Japan	Japan 1W	Japan 250mW
865.7	X	916.8	916.8
866.3	952.4	918.0	918.0
866.9	952.6	919.2	919.2
867.5	952.8	920.4	920.4
	953.0	920.6	920.6
	953.2	920.8	920.8
	953.4		921.0
	953.6		921.2
	953.8		921.4
	X		921.6
			921.8
			922.0
			922.2
			922.4
			922.6
			922.8
			923.0
			923.2
			923.4

uint LBTChannelState{ get; set;}

Value

LBT Channel State 최소:0(0x00) ~ 최대:524287(0x7FFFF)

이 값을 이진수로 변환 했을 때 최하위 비트가 Channel1이 되고 비트가 1이면 해당 채널을 사용한다.

1.8.5 ChannelState

모듈 타입이 MODULE_TYPE_H1000이고 Europe형일 경우에 채널 설정을 위해서 LBTChannelState 대신 ChannelState 속성을 사용할 수도 있다.

uint ChannelState{ get; set;}

Value

Channel State

1.8.6 LBTime

한 채널을 점유하여 Tag를 Read하는 시간을 설정하거나 읽는다. Japan/Europe type에 해당됨. 단위는 ms이며, 전파규정상 최대 점유 시간인 4초를 초과할 수 없다.

`uint LBTime{ get; set; }`

Value

LBT Time 최소:0 ~ 최대:4000

Notes

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.8.7 OperationTime

UHF RFID 장치의 명령 수행시간을 설정하거나 읽는다. 초(sec) 단위이며, 0이면 명령이 수행이 완료 되거나 사용자로부터 Stop 명령을 받을 때 까지 대기한다.

`uint OperationTime{ get; set; }`

Value

Operation Time 최소:0 ~ 최대:60, 기본값:0

1.8.8 PowerLevel

UHF RFID 장치의 Power Level을 설정하거나 읽는다.

`uint PowerLevel{ get; set; }`

Value

Power Level 최소:0 ~ 최대:30, 기본값:0(약 28dBm)

Notes

full power(약 28dBm)에서 감쇄시킬 power level의 값을 기록한다. 예를 들어 PowerLevel = 3 이면 ($28 - 3 = 25$) 설정되는 power level은 25dBm 이다.

1.8.9 ProtocolVersion

UHF RFID 장치의 Protocol Version을 읽는다.

`string` ProtocolVersion{ get; }

Value

Protocol Version

1.8.10 QValue

ReadEpc에 사용되는 Q Parameter 값을 설정하거나 읽는다.

`uint` QValue{ get; set; }

Value

QValue 최소:0 ~ 최대:15, 기본값:4

1.8.11 Session

ReadEpc에 사용되는 Session 값을 설정하거나 읽는다.

`SESSION_TYPE` Session{ get; set; }

Value

Session Type

1.8.12 Selects

SELECT 명령의 실행 횟수.

`int` Selects{ get; set; }

Value

SELECT 실행 횟수.

Notes

xxxEx() 함수로 SELECT 명령을 설정했어도, 이 값이 0이면 SELECT 명령은 실행되지 않는다. SELECT 명령을 설정하지 않았어도 이 값이 1 이상이면 모든 Tag에 대해서 SELECT 명령을 실행한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.8.13 AirDuty

RFID 장치의 TX Cycle을 설정한다.

```
int AirDuty{ set;}
```

Value

Speed 값. (Min:10 ~ Max:100)

값이 커질수록 속도가 증가함.

Notes

배터리 소모, 열 발생 등의 사항과 관련하여 최적의 값이 이미 설정되어 있으므로, 일반적인 용도로는 절대로 이 값을 변경하지 않을 것을 권장한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

1.8.14 ModuleType

PDA에 장착된 RFID 모듈의 타입을 리턴한다.

```
uint ModuleType{ get; }
```

Value

RFID 모듈 타입

Notes

RFID 모듈 타입에 따라 일부 기능에 차이가 있다.

1.2 Constants 참조.

1.8.15 MaxPwr

PDA에 장착된 RFID 모듈의 최대 Tx Power 값을 리턴한다.

```
uint MaxPwr{ get; }
```

Value

30x : 1W

28x : 600mW

27x : 500mW

24x : 250mW

1.8.16 EnableTagFocus

MODULE_TYPE_I2000에서만 동작하는 기능으로,

Impinj Extension(Tag Focus) 기능을 Enable/Disable 한다.

EnableTagFocus 속성이 True로 설정되어 있는 동안은 InventoryTarget, Session 속성을 변경할 수 없다.

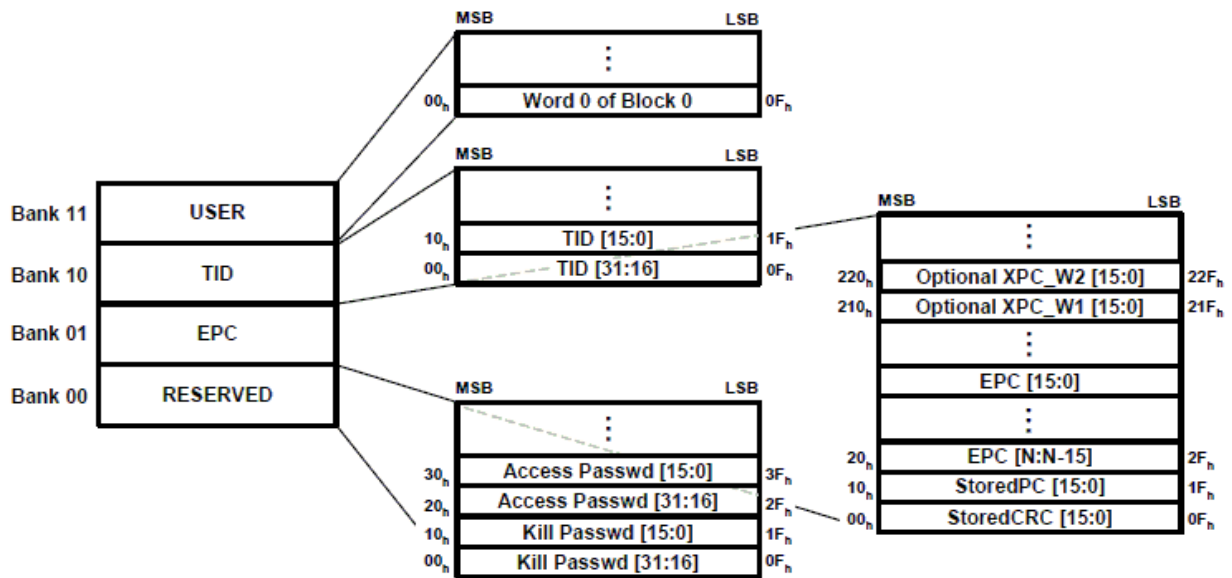
`bool` EnableTagFocus{ get; set;}

Value

Enable/Disable 상태

2 C/C++ API Reference

- ISO 18000 6C(EPC Class 1 Generation 2) Logical Memory Map



2.1 Enumerations

2.1.1 RFID_RESULT

함수의 호출 결과를 나타낸다.

- **RFID_RESULT_SUCCESS**

기능 수행 성공.

- **RFID_RESULT_OUTOFMEMORY**

자원 할당 실패.

- **RFID_RESULT_INVALID_ARGS**

유효하지 않은 parameter.

- **RFID_RESULT_FAILURE**

기능 수행 실패.

- **RFID_RESULT_UNEXPECTED**

알 수 없는 Error 발생.

- **RFID_RESULT_ALREADY_OPENED**

UHF RFID 장치 port가 이미 열려 있음.

- **RFID_RESULT_INVALID_DEVICE**

지원하지 않는 장치 타입.

- **RFID_RESULT_NOT_CONNECTED**

장치 open 후 정상적으로 반응 하지 않음.

- **RFID_RESULT_NOT_DETECT**

Tag를 찾지 못하고 명령이 종료됨.

- **RFID_RESULT_ACCESS_ERROR**

잘못된 패스워드로 접근 시도 또는 존재 하지 않는 영역에 접근 시도.

- **RFID_RESULT_NOT_OPENED**

Open을 호출하지 않고 기능 수행을 시도 함.

- **RFID_RESULT_COMMAND_ERROR**

명령 실행 중 에러 발생, 또는 명령 실행이 종료 되지 않은 상태에서 다른 명령을 받음.

- **RFID_RESULT_LOW_BATTERY**

모듈에 공급되는 전원의 전압이 낮아 동작 할 수 없음.

- **RFID_RESULT_UNKNOWN**

알 수 없는 Error 발생.

- **RFID_RESULT_NOT_SUPPORTED**

지원 되지 않는 명령.

- **RFID_RESULT_STOPPED**

명령 실행이 중단 됨.

- **RFID_RESULT_POWER_OFF**

UHF RFID 장치의 전원이 off되어 있음.

2.1.2 **RFIDMEM_BANK**

RFID 장치가 access할 Tag의 Memory bank

- **RFID_MEM_BANK_RESERVED**

kill password 또는 access password가 저장되는 memory.

- **RFID_MEM_BANK_EPC**

StoredCRC, EPC, XPC 정보들이 저장되는 memory.

- **RFID_MEM_BANK_TID**

ISO/IEC 15963 allocation class identifier, Tag 일련 번호 등이 저장된다.

- **RFID_MEM_BANK_USER**

user specific data storage.

2.1.3 **RFIDREAD_TYPE**

Tag reading(Inventory) 시 적용될 single, multiple mode.

- **RFID_EPC_GEN2_MULTI_TAG**

다수의 Tag를 동시에 읽는다.

- **RFID_EPC_GEN2_ONE_TAG**

한 장의 Tag만 읽고 읽기를 중단 한다.

2.1.4 **RFIDPERMALOCK_FIELD**

Permalock을 적용 시킬 field

- **RFID_PERMALOCK_FIELD_ACCESS_PASSWORD**

Tag의 Access password field

- **RFID_PERMALOCK_FIELD_KILL_PASSWORD**

Tag의 Kill password field

- **RFID_PERMALOCK_FIELD_EPC**

Tag의 EPC bank

- **RFID_PERMALOCK_FIELD_TID**

Tag의 TID bank

- **RFID_PERMALOCK_FIELD_USER**

Tag의 USER bank

2.1.5 RFIDCALLBACKTYPE

Callback delegate가 실행되는 이유를 나타내는 값

- **RFIDCALLBACKTYPE_DATA**

Tag로부터 Memory data를 읽었기 때문에 발생함.

- **RFIDCALLBACKTYPE_REPLY**

RFID 장치로부터 명령의 실행 결과를 받았을 때 발생함.

2.1.6 RFIDINVENTORIEDSTATE

ReadEpc(Inventory)를 수행할 때 Target으로 사용할 Tag의 Inventoried state.

- **RFID_INVENTORIEDSTATE_A**

Inventoried state가 A 인 Tag만 Inventory 함.

- **RFID_INVENTORIEDSTATE_B**

Inventoried state가 B 인 Tag만 Inventory 함.

- **RFID_INVENTORIEDSTATE_AB**

Inventoried state가 A또는B 인 Tag를 Inventory 함.

2.1.7 RFIDSESSIONTYPE

ReadEpc(Inventory)를 수행할 때 사용할 Session

- **RFID_SESSIONTYPE_0**

S0

- **RFID_SESSIONTYPE _1**

S1

- **RFID_SESSIONTYPE _2**

S2

- **RFID_SESSIONTYPE _3**

S3

2.2 Constants

2.2.1 WM_RFID_RESPONSE

WM_COPYDATA 메시지가 발생 했을 때 RFID의 결과 임을 알리는 값.

COPYDATASTRUCT의 dwData 값이 WM_RFID_RESPONSE와 같다면 RFID에 대한 결과가 도착한 것이다.

```
#define WM_RFID_RESPONSE WM_USER + 1804
```

2.2.2 MAX_MASKS_BYTES

Masking pattern으로 사용할 수 있는 최대 바이트 수

```
- #define MAX_MASKS_BYTES 32
```

2.2.3 MODULE_TYPE_NONE

RFID 모듈이 장착되어 있지 않음.

```
- #define MODULE_TYPE_H1000 0x30
```

2.2.4 MODULE_TYPE_H1000

2013년 1월 현재, 일본을 제외한 국가에 해당하는 RFID 모듈

```
#define MODULE_TYPE_H1000 0x31
```

2.2.5 MODULE_TYPE_I2000

2013년 1월 현재, 일본에만 해당하는 RFID 모듈

일본의 변경된 전파법이 적용됨.

```
#define MODULE_TYPE_I2000 0x32
```

2.2.6 MODULE_TYPE_I900

2013년 1월 현재, AT980에만 장착되는 RFID 모듈

```
#define MODULE_TYPE_I900 0x33
```


2.3 Reply Words List

SyncMode를 사용하지 않고 Tag Operation 함수들을 실행하면 Callback delegate를 통해서 아래와 같은 문자열을 받는다.

- **“Other Error”**

알 수 없는 에러 발생.

- **“Memory Overrun”**

존재 하지 않는 Memory 영역에 접근을 시도 함.

- **“Memory Locked”**

Tag가 잠겨 있음.

- **“Insufficient Tag Power”**

Tag의 신호가 약해서 기능 수행 실패.

- **“Non-specific Error”**

알 수 없는 에러 발생.

- **“Check Antenna”**

RFID 장치와 Antenna간의 연결 상태 불량.

- **“Try after cooled”**

RFID 장치가 과열 됨.

- **“Insufficient PDA Power”**

PDA 메인 배터리 전원이 낮아서 기능 수행 실패.

- **“Not Supported”**

지원되지 않는 기능.

- **“Not Connected”**

RFID 장치가 open 이후 정상적으로 반응하지 않음.

- **“Not Opened”**

Open 함수를 수행하지 않고 기능 함수를 호출함.

- **“Bad Access Password”**

잘못된 Access Password로 Tag에 접근 함.

- **“Invalid Parameter”**

유효하지 않은 parameter

- **“Command Error”**

명령 수행 도중 에러 발생 또는 이전 명령이 종료 되기 전에 또 다른 명령을 실행 함.

- **“Success”**

성공적으로 함수 실행 완료.

- **“Not Detect”**

Tag를 찾지 못하고 명령이 종료 됨.

- **“Multi Read Stop”**

ReadEpc(multiple) 명령이 Stop 명령에 의해서 종료 됨.

- **“EAS”**

NXP Tag의 EAS가 감지 됨.

2.4 Structures

2.4.1 RFIDLOCK_UNLOCK_FIELD

Tag의 각 field를 lock 또는 unlock 하기 위한 정보를 저장한다.
True로 설정된 멤버에 대해서만 lock 또는 unlock이 적용된다.

```
typedef struct
{
    bool bAccessPassword;
    bool bEPC;
    bool bKillPassword;
    bool bTID;
    bool bUSER;
} RFIDLOCK_UNLOCK_FIELD, *RFIDLOCK_UNLOCK_FIELD;
```

- **bAccessPassword**

Tag의 access password field

- **bEPC**

Tag의 EPC bank

- **bKillPassword**

Tag의 kill password field

- **bTID**

Tag의 TID bank

- **bUSER**

Tag의 USER bank.

Ex) Access Password와 EPC를 Lock 또는 Unlock하기 위해서는 아래와 같이 구성한다.

RFIDLOCK_UNLOCK_FIELD Lock;

Lock.bAccessPassword = true;

Lock. bEPC = true;

Lock. bKillPassword = false;

Lock. bTID = false;

Lock. bUSER = false;

2.4.2 RFIDMASK_PARAMS

태그에 대한 작업을 수행하는 명령 호출 시에 조건에 해당되는 태그에만 반응 하도록 할 때 사용한다.

```
typedef struct
{
    BYTE MemBank;
    UINT nOffSet;
    LPWSTR szMaskPattern;
} RFIDMASK_PARAMS, * RFIDPMASK_PARAMS;
```

- **MemBank**

Mask로 사용될 태그의 Memory Bank

- **nOffSet**

선택한 Memory Bank에서 MaskPattern의 nibble offset

- **szMaskPattern**

4bit로 지정된 masking pattern string

Ex) TID bank가 "E2006004015F325A"로 입력되어 있는 태그를 TID bank를 Mask로 사용하여 읽기 위해서는 아래와 같이 구성할 수 있다..

RFIDMASK_PARAMS Mask;

Mask.MemBank = MEM_BANK.TID;

Mask.nOffSet = 1;

Mask.szMaskPattern = _T("2006");

또는

RFIDMASK_PARAMS Mask;

Mask.MemBank = MEM_BANK.TID;

Mask.nOffSet = 11;

Mask.szMaskPattern = _T("F325");

2.4.3 RFIDMASK_PARAMS_EX

Masking pattern을 bit 단위로 적용할 때 사용한다.

```
typedef struct
{
    BYTE MemBank;
    UINT nOffSet;
    UINT nBits;
    BYTE byMaskPattern[MAX_MASK_BYTES];
} RFIDMASK_PARAMS_EX, * RFIDPMASK_PARAMS_EX;
```

- **MemBank**

Mask로 사용될 태그의 Memory Bank

- **nOffSet**

선택한 Memory Bank에서 byMaskPattern의 bit offset

- **nBits**

masking pattern으로 사용될 byMaskPattern의 bit 수.

- **byMaskPattern**

masking pattern byte array

2.4.4 RFID_MASK_PARAMS_EX2

Bit 단위의 Masking pattern을 사용하여 SELECT 명령을 적용할 때 사용된다.

typedef struct

```
{
    BYTE ActionCode;
    UINT mskBits;
    BYTE mskPattern[MAX_MASK_BYTES];
    BYTE mskMemBank;
    UINT mskOffSet;
    BYTE SelectTarget;
} RFID_MASK_PARAMS_EX2, * PRFID_MASK_PARAMS_EX2;
```

- **ActionCode**

Tag의 flag에 적용할 Select command

- **mskBits**

masking pattern으로 사용될 MaskPattern의 bit 수.

- **mskPattern**

masking pattern byte array

- **mskMemBank**

Mask로 사용될 태그의 Memory Bank

- **mskOffSet**

선택한 Memory Bank에서 MaskPattern의 bit offset

- **SelectTarget**

Select command가 적용될 Tag의 flag

ActionCode, SelectTarget의 조합으로 아래 표와 같이 Tag의 상태를 변경 할 수 있다.

	If SELECT_TARGET is 4(SL)		If SELECT_TARGET are between 0 ~ 3	
	Matching	Non-matching	Matching	Non-matching
ActionCode is 0	Assert SL	De-assert SL	→ A	→ B
ActionCode is 1	Assert SL	No Action	→ A	No Action
ActionCode is 2	No Action	De-assert SL	No Action	→ B
ActionCode is 3	Negate SL	No Action	Negate	No Action

ActionCode is 4	De-assert SL	Assert SL	→ B	→ A
ActionCode is 5	De-assert SL	No Action	→ B	No Action
ActionCode is 6	No Action	Assert SL	No Action	→ A
ActionCode is 7	No Action	Negate SL	No Action	Negate

2.4.5 RFIDCALLBACKDATA

RFID 장치로부터 응답을 받았을 때 callback function을 통해서 받는 구조체.

Callback function이 호출되면 이 구조체의 멤버들을 가지고 RfidGetResult 함수를 사용하여 데이터를 읽는다.

typedef struct

```
{
    RFIDCALLBACKTYPE CallbackType;
    LPARAM IParam
    WPARAM wParam;
} RFIDCALLBACKDATA;
```

- CallbackType

Callback function 실행의 원인.(Data or Reply)

반드시 RfidGetResult 함수의 파라미터로 전달해야 한다.

- IParam

RFID 장치로부터 데이터를 읽어 오기 위한 파라미터 1.

반드시 RfidGetResult 함수의 파라미터로 전달해야 한다.

- wParam

RFID 장치로부터 데이터를 읽어 오기 위한 파라미터 2

반드시 RfidGetResult 함수의 파라미터로 전달해야 한다.

2.4.6 RFIDAIRDURAPARAMS

모듈의 Tx Cycle 동작 시간을 조정할 때 사용하는 구조체.

typedef struct

```
{
    UINT nOnMs;
    UINT nOffMs;
} RFIDAIRDURAPARAMS;
```

- nOnMs

Tx Cycle에서의 동작 시간.(millisecond)

Min:0ms ~ Max:4000ms

- nOffMs

Tx Cycle에서의 쉬는 시간.(millisecond)

Min:0ms ~ Max:4000ms

2.5 Callback function definition

2.5.1 RFIDCALLBACK

RFID 장치로부터 응답을 받았을 때 실행되는 callback function

```
typedef void (CALLBACK* RFIDCALLBACK)(RFIDCALLBACKDATA* pCallbackData);
```

- pCallbackData

RFID 장치로부터 받은 응답이 들어 있는 구조체.

2.6 Methods

2.6.1 RfidPowerOn

UHF RFID 장치의 전원을 인가한다.

```
RFID_RESULT RfidPowerOn();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

UHF RFID 장치를 사용하기 위해서는 반드시 호출해야 한다.

2.6.2 RfidPowerOff

UHF RFID 장치에 인가된 전원을 제거한다.

```
void RfidPowerOff();
```

Parameters

None

Return Values

None

Notes

UHF RFID 장치 사용이 끝나면, 이 함수를 호출하여 전원을 해제 한다.

2.6.3 RfidOpen

시스템 자원을 할당하고, UHF RFID 장치의 port를 open 한다.

```
RFID_RESULT RfidOpen();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.4 RfidClose

UHF RFID 장치의 port를 close하고, 할당된 자원을 해제한다.

```
void Close();
```

Parameters

None

Return Values

None

2.6.5 RfidIsOpened

UHF RFID 장치의 port가 open되어 있는지 확인한다.

```
BOOL RfidIsOpened()
```

Parameters

None

Return Values

UHF RFID 장치의 open 상태.

True:Open, False:Closed

2.6.6 RfidSetCallback

RFID 장치에서 응답을 받았을 때 호출될 callback function을 등록한다.

```
RFID_RESULT RfidSetCallback (  
    RFIDCALLBACK pCallback  
);
```

Parameters

pCallback

RFIDCALLBACK callback function.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

RfidOpen() 을 호출한 이후에 호출해야 한다.

2.6.7 RfidSetHwnd

Callback function 대신 window message를 통지 받아 결과를 처리하고자 할 때 application windows handle을 설정한다.

```
RFID_RESULT RfidSetHwnd (
    HWND hWnd
);
```

Parameters

hWnd

application windows handle

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

RfidOpen() 을 호출한 이후에 호출해야 한다.

2.6.8 RfidGetResult

Tag Data 또는 RFID 장치로부터의 응답을 읽는다.

```
RFID_RESULT RfidGetResult (
    LPWSTR szResult,
    RFIDCALLBACKTYPE CallbackType,
    WPARAM wParam,
    LPARAM lParam
);
```

Parameters

szResult

Tag로부터 읽은 Data 또는 Reply가 저장될 변수.

CallbackType

Callback function이 실행될 때 넘겨 받은 RFIDCALLBACKDATA의 CallbackType을 그대로 CallbackType parameter 값으로 설정해준다. 또한 응용프로그램에서는 RFIDCALLBACKDATA의 CallbackType의 내용을 보고 callback function의 호출이 data 또는 응답(reply)인지 파악하고 해당 type에 맞게 처리한다.

wParam

Callback function이 호출될 때 parameter로 전달 받은 RFIDCALLBACKDATA의 wParam을 그대로 설정해준다.

lParam

Callback function이 호출될 때 parameter로 전달 받은 RFIDCALLBACKDATA의 IParam을 그대로 설정해준다.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

이 함수는 반드시 사용자가 등록한 callback function 내에서 호출해야 한다.

2.6.9 RfidReadEpc

Tag로부터 EPC 데이터를 읽는다. (Inventory)

```
RFID_RESULT RfidReadEpc(
    BOOL bSyncMode,
    RFIDREAD_TYPE ReadType,
    RFIDPMASK_PARAMS pMask,
    LPWSTR szEpcData
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

ReadType

Singe Tag Reading 또는 Multiple Tag Reading 모드.

pMask

Inventory에 사용될 masking pattern. (nibble단위의 masking pattern)

szEpcData

Sync 모드로 호출 했을 때, EPC 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.10 RfidReadEpcEx

Tag로부터 EPC 데이터를 읽는다. (Inventory)

bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidReadEpcEx(
    BOOL bSyncMode,
    RFIDREAD_TYPE ReadType,
    RFIDPMASK_PARAMS_EX pMask,
    LPWSTR szEpcData
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

ReadType

Singe Tag Reading 또는 Multiple Tag Reading 모드.

pMask

Inventory에 사용될 bit단위의 masking pattern.

EpcData

Sync 모드로 호출 했을 때, EPC 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.11 RfidReadEpcEx2

Tag로부터 EPC 데이터를 읽는다. (Inventory)

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidReadEpcEx2(
    BOOL bSyncMode,
    RFIDREAD_TYPE ReadType,
    PRFID_MASK_PARAMS_EX2 pMasks,
    UINT nNumberOfMasks,
    LPWSTR szEpcData
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

ReadType

Singe Tag Reading 또는 Multiple Tag Reading 모드.

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

EpcData

Sync 모드로 호출 했을 때, EPC 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.12 RfidStop

UHF RFID 장치에서 Async 모드로 수행중인 작업을 중단 시킨다.

```
RFID_RESULT RfidStop();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.13 RfidStopEx

UHF RFID 장치에서 Async 모드로 수행중인 작업을 중단 시킨다.

```
RFID_RESULT RfidStop(  
    BOOL block  
);
```

Parameters

Block

동작 모드

TRUE : stop operation이 종료될 때까지 대기하고 리턴된다.(최대 5초)

FALSE : stop 명령을 전송하고 함수가 즉시 리턴된다.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.14 RfidReadMemBank

Tag의 특정 memory bank로부터 word(2bytes) 단위로 data를 읽는다.

```
RFID_RESULT RfidReadMemBank(  
    BOOL bSyncMode,  
    RFIDMEM_BANK MemBank,  
    UINT nWordPtr,  
    UINT nWordCount,  
    BOOL bContinuous,
```

```

    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS pMask,
    LPWSTR szMemBankData
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

읽을 Memory Bank Type

nWordPtr

읽을 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

nWordCount

읽을 data 크기. (Min:0 ~ Max:255)

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

pMask

nibble 단위의 masking pattern.

szMemBankData

Sync 모드로 호출 했을 때, MemBank 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.15 RfidReadMemBankEx

Tag의 특정 memory bank로부터 word(2bytes) 단위로 data를 읽는다.
bit단위의 MASK 기능을 지원한다.

```

RFID_RESULT RfidReadMemBankEx(
    BOOL bSyncMode,
    RFIDMEM_BANK MemBank,
    UINT nWordPtr,
    UINT nWordCount,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARMAS_EX pMask,

```

LPWSTR szMemBankData

);

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

읽을 Memory Bank Type

nWordPtr

읽을 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

nWordCount

읽을 data 크기. (Min:0 ~ Max:255)

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

pMask

bit 단위의 masking pattern.

szMemBankData

Sync 모드로 호출 했을 때, MemBank 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.16 RfidReadMemBankEx2

Tag의 특정 memory bank로부터 word(2bytes) 단위로 data를 읽는다.

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidReadMemBankEx2(
    BOOL bSyncMode,
    RFIDMEM_BANK MemBank,
    UINT nWordPtr,
    UINT nWordCount,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    PRFID_MASK_PARMAS_EX2 pMasks,
    UINT nNumberOfMasks,
    LPWSTR szMemBankData
```

);

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

읽을 Memory Bank Type

nWordPtr

읽을 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

nWordCount

읽을 data 크기. (Min:0 ~ Max:255)

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

szMemBankData

Sync 모드로 호출 했을 때, MemBank 값이 저장될 변수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.17 RfidWriteMemBank

Tag의 특정 memory bank에 word(2bytes) 단위로 data를 기록한다.

RFID_RESULT RfidWriteMemBank (

BOOL bSyncMode,

RFIDMEM_BANK MemBank,

UINT nWordPtr,

LPWSTR szData,

BOOL bContinuous,

LPWSTR szAccessPassword,

RFIDPMASK_PARAMS pMask,

);

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

데이터를 기록할 Memory Bank Type

nWordPtr

데이터를 기록할 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

szData

기록할 데이터.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

pMask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.18 RfidWriteMemBankEx

Tag의 특정 memory bank에 word(2bytes) 단위로 data를 기록한다.
bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidWriteMemBankEx (
    BOOL bSyncMode,
    RFIDMEM_BANK MemBank,
    UINT nWordPtr,
    LPWSTR szData,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS_EX pMask
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

데이터를 기록할 Memory Bank Type

nWordPtr

데이터를 기록할 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

szData

기록할 데이터.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

pMask

bit 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.19 RfidWriteMemBankEx2

Tag의 특정 memory bank에 word(2bytes) 단위로 data를 기록한다.

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidWriteMemBankEx2 (
    BOOL bSyncMode,
    RFIDMEM_BANK MemBank,
    UINT nWordPtr,
    LPWSTR szData,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    PRFID_MASK_PARAMS_EX2 pMasks,
    UINT nNumberOfMasks
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

MemBank

데이터를 기록할 Memory Bank Type

nWordPtr

데이터를 기록할 Memory Bank 내의 시작 offset. (Min:0 ~ Max: 422820625)

szData

기록할 데이터.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

memory bank에 lock이 걸려 있을 경우 사용할 access password.

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.20 RfidLockField

Tag의 특정 field를 lock 한다. Lock이 된 field 접근시에는 Access password를 필요로 한다.

```
RFID_RESULT RfidLockField (
    BOOL bSyncMode,
    RFIDPLOCK_UNLOCK_FIELD pLockField,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS pMask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

pLockField

Lock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

AccessPassword

access password.

pMask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.21 RfidLockFieldEx

Tag의 특정 field를 lock 한다. Lock이 된 field 접근시에는 Access password를 필요로 한다.

bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidLockFieldEx (
    BOOL bSyncMode,
    RFIDPLOCK_UNLOCK_FIELD pLockField,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS_EX pMask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

pLockField

Lock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMask

bit 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.22 RfidLockFieldEx2

Tag의 특정 field를 lock 한다. Lock이 된 field 접근시에는 Access password를 필요로 한다.

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidLockFieldEx2 (
```

```

    BOOL bSyncMode,
    RFIDPLOCK_UNLOCK_FIELD pLockField,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    PRFID_MASK_PARAMS_EX2 pMasks,
    UINT nNumberOfMasks
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

pLockField

Lock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.23 RfidUnLockField

Tag의 특정 field를 unlock 한다.

```

RFID_RESULT RfidUnLockField (
    BOOL bSyncMode,
    RFIDPLOCK_UNLOCK_FIELD pLockField,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS pMask,
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

pLockField

UnLock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.24 RfidUnLockFieldEx

Tag의 특정 field를 lock 한다

bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidUnLockFiledEx (
    BOOL bSyncMode,
    RFIDPLOCK_UNLOCK_FIELD pLockField,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS_EX pMask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

pLockField

UnLock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMask

bit 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.25 RfidUnLockFieldEx2

Tag의 특정 field를 lock 한다

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidUnLockFiledEx2 (
    BOOL bSyncMode,
    RFIDPLOCK_UNLOCK_FIELD pLockField,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    PRFID_MASK_PARAMS_EX2 pMasks,
    UINT nNumberOfMasks
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

pLockField

UnLock을 적용할 field

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.26 RfidPermalock

Tag의 특정 field를 Permalock 한다.

영구적으로 lock이 된 field는 unlock으로 secured 상태를 해제할 수 없다. Secured 상태는 tag의 password가 enable 되어 있어 tag 접근 시 password를 필요로 하는 상태를 의

미한다.

```
RFID_RESULT RfidPermalock (
    BOOL bSyncMode,
    RFIDPERMALOCK_FIELD PermalockField,
    BOOL bSecured,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS pMask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

PermalockField

Permalock을 적용할 field

bSecured

Secured 상태.

True로 설정되면 영구히 access password를 가지고 Tag에 access 해야한다.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 secured 상태를 영구히 변경할 수 없으므로 주의해야 한다.

2.6.27 RfidPermalockEx

Tag의 특정 field를 permalock 한다

bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidPermalockEx (
    BOOL bSyncMode,
    RFIDPERMALOCK_FIELD PermalockField,
```



```

    BOOL bSecured,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    RFIDPMASK_PARAMS_EX pMask,
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

LockField

Permalock을 적용할 field

bSecured

Secured 상태.

True로 설정되면 영구히 access password를 가지고 Tag에 access 해야한다.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMask

bit 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

2.6.28 RfidPermalockEx2

Tag의 특정 field를 permalock 한다

SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```

RFID_RESULT RfidPermalockEx2 (
    BOOL bSyncMode,
    RFIDPERMALOCK_FIELD PermalockField,
    BOOL bSecured,
    BOOL bContinuous,
    LPWSTR szAccessPassword,
    PRFID_MASK_PARAMS_EX2 pMask,
    UINT nNumberOfMasks
);

```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

LockField

Permalock을 적용할 field

bSecured

Secured 상태.

True로 설정되면 영구히 access password를 가지고 Tag에 access 해야한다.

bContinuous

Continuous 모드. (True=Enable, False=Disable)

szAccessPassword

access password.

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.29 RfidKillTag

Tag를 UHF RFID 장치에 반응하지 않도록 Kill한다. Kill password가 "00000000"인 경우에는 kill을 할 수 없다. Kill 명령을 수행하기 위해 tag는 "00000000"가 아닌 다른 password가 설정이 되어 있어야 한다. 한 번 killed 된 tag는 다시 복구할 수 없다.

```
RFID_RESULT RfidKillTag (
    BOOL bSyncMode,
    LPWSTR szKillPassword,
    BOOL bContinuous,
    RFIDPMASK_PARAMS pMask
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

szKillPassword

Tag에 저장되어 있는 Kill Password

bContinuous

Continuous 모드. (True=Enable, False=Disable)

Mask

nibble 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 다시 복구 할 수 없으므로 주의해야 한다.

2.6.30 RfidKillTagEx

Tag를 UHF RFID 장치에 반응하지 않도록 Kill한다. Kill password가 "00000000"인 경우에는 kill을 할 수 없다. Kill 명령을 수행하기 위해 tag는 "00000000"가 아닌 다른 password가 설정이 되어 있어야 한다. 한 번 killed 된 tag는 다시 복구할 수 없다.
bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidKillTagEx (
    BOOL bSyncMode,
    LPWSTR szKillPassword,
    BOOL bContinuous,
    RFIDPMASK_PARAMS_EX pMask,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

szKillPassword

Tag에 저장되어 있는 Kill Password

bContinuous

Continuous 모드. (True=Enable, False=Disable)

pMask

bit 단위의 masking pattern.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

Sync 모드에서는 Continuous 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 다시 복구 할 수 없으므로 주의해야 한다.

2.6.31 RfidKillTagEx2

Tag를 UHF RFID 장치에 반응하지 않도록 Kill한다. Kill password가 "00000000"인 경우에

는 kill을 할 수 없다. Kill 명령을 수행하기 위해 tag는 "00000000"가 아닌 다른 password가 설정이 되어 있어야 한다. 한 번 killed 된 tag는 다시 복구할 수 없다.
SELECT 기능과 bit단위의 MASK 기능을 지원한다.

```
RFID_RESULT RfidKillTagEx2 (
    BOOL bSyncMode,
    LPWSTR szKillPassword,
    BOOL bContinuous,
    PRFID_MASK_PARAMS_EX2 pMasks,
);
```

Parameters

bSyncMode

함수 동작 모드. (True=Sync, False=Async)

szKillPassword

Tag에 저장되어 있는 Kill Password

bContinuous

Continuous 모드. (True=Enable, False=Disable)

pMasks

bit단위의 masking pattern의 주소.

nNumberOfMasks

pMasks의 데이터 개수.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

pMasks를 사용하는 경우에는 Sync 모드를 지원하지 않는다.

이 함수가 성공적으로 완료 되면 다시 복구 할 수 없으므로 주의해야 한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.32 RfidSetDefault

UHF RFID 장치의 설정 값들을 초기값으로 되돌린다.

```
RFID_RESULT RfidSetDefault ();
```

Parameters

None

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.33 RfidSetExtInfoEnable

Tag를 읽을 때 RSSI 값을 부가적으로 읽도록 설정하거나 해제한다.

```
RFID_RESULT RfidSetExtInfoEnable (
    BOOL bEnable
);
```

Parameters

bEnable

Extended information(RSSI) 설정 여부. (True:On, False:Off)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.34 RfidGetQValue

ReadEpc에 사용되는 Q Parameter 값을 읽는다.

```
RFID_RESULT RfidGetQValue (
    UINT* pnQvalue
);
```

Parameters

pnQvalue

Q Paramertet 값. (Min:0 ~ Max:15)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.35 RfidSetQValue

ReadEpc에 사용되는 Q Parameter 값을 설정한다.

```
RFID_RESULT RfidSetQValue (
    UINT nQvalue
);
```

Parameters

nQvalue

Q Paramertet 값. (Min:0 ~ Max:15)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.36 RfidGetOperationTime

RFID 장치의 명령 수행시간을 읽는다. 0이면 명령이 수행 완료 되거나 사용자로부터 Stop 명령을 받을 때 까지 대기한다.

```
RFID_RESULT RfidGetOperationTime (  
    UINT* pnOperationTime  
);
```

Parameters

pnOperationTime

명령 수행 시간. (Min:0 ~ Max:60)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.37 RfidSetOperationTime

RFID 장치의 명령 수행시간을 설정한다. 0이면 명령이 수행 완료 되거나 사용자로부터 Stop 명령을 받을 때 까지 대기한다.

```
RFID_RESULT RfidSetOperation (  
    UINT nOperationTime  
);
```

Parameters

nOperationTime

명령 수행 시간. (Min:0 ~ Max:60)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.38 RfidGetPowerLevel

RFID 장치의 Power level을 읽는다.

```
RFID_RESULT RfidGetPowerLevel (  
    UINT* pnPowerLevel  
);
```

Parameters

pnPowerLevel

Power level (Min:0 ~ Max:30) 기본값(약 28dBm)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.39 RfidSetPowerLevel

RFID 장치의 Power level을 설정한다.

```
RFID_RESULT RfidSetPowerLevel (
    UINT nPowerLevel
);
```

Parameters

nPowerLevel

Power level (Min:0 ~ Max:30) 기본값(약 28dBm)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.40 RfidGetSession

ReadEpc에 사용될 Session 값을 읽는다.

```
RFID_RESULT RfidGetSession (
    RFIDSESSIONTYPE* pSessionType
);
```

Parameters

pSessionType

Session

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.41 RfidSetSession

ReadEpc에 사용될 Session 값을 설정한다.

```
RFID_RESULT RfidSetSession (
    RFIDSESSIONTYPE SessionType
);
```

);

Parameters

SessionType

Session

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.42 RfidGetInventoryTarget

ReadEpc에 사용될 Target 값을 읽는다.

```
RFID_RESULT RfidGetInventoryTarget (
    RFIDINVENTORIEDSTATE* pInventoriedState
);
```

Parameters

pInventoriedState

Inventory Target

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.43 RfidSetInventoryTarget

ReadEpc에 사용될 Target 값을 설정한다.

```
RFID_RESULT RfidSetInventoryTarget (
    RFIDINVENTORIEDSTATE InventoriedState
);
```

Parameters

InventoriedState

Inventory Target

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.44 RfidGetLbtChState

LBT Channel을 읽는다. Japan, Europe type에 해당됨.

모듈별 사용 주파수와 채널 수는 아래의 표와 같다.

MODULE_TYPE_H1000		MODULE_TYPE_I2000	
Europe	Japan	Japan 1W	Japan 250mW
865.7	X	916.8	916.8
866.3	952.4	918.0	918.0
866.9	952.6	919.2	919.2
867.5	952.8	920.4	920.4
	953.0	920.6	920.6
	953.2	920.8	920.8
	953.4		921.0
	953.6		921.2
	953.8		921.4
	X		921.6
			921.8
			922.0
			922.2
			922.4
			922.6
			922.8
			923.0
			923.2
			923.4

```
RFID_RESULT RfidGetLbtChState (
    UINT* pnLbtChState
);
```

Parameters

pnLbtChState

LBT Channel State 최소:0(0x00) ~ 최대:524287(0x7FFFF)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

이 값을 이진수로 변환 했을 때 최하위 비트가 Channel1이 되고 비트가 1이면 해당 채널을 사용한다.

2.6.45 RfidSetLbtChState

LBT Channel state 값을 변경한다. Japan/Europe type에 해당됨.

2.6.44 RfidGetLbtChState 표 참조.

```
RFID_RESULT RfidSetLbtChState (
    UINT nLbtChState
);
```

Parameters

nLbtChState

LBT Channel State 최소:0(0x00) ~ 최대:524287(0x7FFFF)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

이 값을 이진수로 변환 했을 때 최하위 비트가 Channel1이 되고 비트가 1이면 해당 채널을 사용한다.

2.6.46 RfidGetChannelState

모듈 타입이 MODULE_TYPE_H1000이고 Europe형일 경우에 채널 설정을 위해서 RfidGetLbtChState 대신 RfidGetChannelState 함수를 사용할 수도 있다.

```
RFID_RESULT RfidGetChannelState (
    UINT* pnChState
);
```

Parameters

pnChState

Channel State

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

이 값을 이진수로 변환 했을 때 최하위 비트가 Channel1이 되고 비트가 1이면 해당 채널을 사용한다.

2.6.47 RfidSetChannelState

모듈 타입이 MODULE_TYPE_H1000이고 Europe형일 경우에 채널 설정을 위해서 RfidSetLbtChState 대신 RfidSetChannelState 함수를 사용할 수도 있다.

```
RFID_RESULT RfidSetChannelState (
    UINT nChState
);
```

Parameters

nChState
Channel State

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

이 값을 이진수로 변환 했을 때 최하위 비트가 Channel1이 되고 비트가 1이면 해당 채널을 사용한다.

2.6.48 RfidGetLbtTime

LBT Channel 을 점유하여 Tag를 Read하는 시간 값을 읽는다.. Japan/Europe type에 해당됨. 단위는 ms 이며, 전파규정상 최대 점유 시간인 4초를 초과할 수 없다.

```
RFID_RESULT RfidGetLbtTime (
    UINT* pnLbtTime
);
```

Parameters

pnLbtTime
LBT Time (Min:0 ~ Max:4000)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.49 RfidSetLbtTime

LBT Channel 을 점유하여 Tag를 Read하는 시간 값을 설정한다.. Japan/Europe type에 해당됨. 단위는 ms 이며, 전파규정상 최대 점유 시간인 4초를 초과할 수 없다.

```
RFID_RESULT RfidSetLbtTime (
    UINT nLbtTime
);
```

Parameters

nLbtTime

LBT Time (Min:0 ~ Max:4000)

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.50 RfidSetAirDuty

RFID 장치의 TX Cycle을 설정한다.

```
RFID_RESULT RfidSetAirDuty (
    UINT nSpeed
);
```

Parameters

nSpeed

Speed 값. (Min:10 ~ Max:100)

값이 커질수록 속도가 증가함.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

Notes

배터리 소모, 열 발생 등의 사항과 관련하여 최적의 값이 이미 설정되어 있으므로, 일반적인 용도로는 절대로 이 값을 변경하지 않을 것을 권장한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.51 RfidGetFirmwareVersion

RFID 장치의 firmware version을 읽는다.

```
RFID_RESULT RfidGetFirmwareVersion (
    LPWSTR szVersion
);
```

Parameters

szVersion

firmware version

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.52 RfidGetModuleType

PDA에 장착된 RFID 모듈의 타입을 리턴한다.

```
uint RfidGetModuleType ( );
```

Parameters

None

Return Values

RFID 모듈 타입에 따라 일부 기능에 차이가 있다.

2.2 Constants 참조.

2.6.53 RfidGetXcvrMaxPwr

PDA에 장착된 RFID 모듈의 최대 Tx Power 값을 리턴한다.

```
RFID_RESULT RfidGetXcvrMaxPwr (
    UINT * pPwr
);
```

Parameters

pPwr

30x : 1W

28x : 600mW

27x : 500mW

24x : 250mW

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 반환한다.

2.6.54 RfidIsRunning

Rfid 모듈이 Operation 진행 중인지 여부를 반환한다.

```
BOOL IsRunning ( );
```

Parameters

None

Return Values

모듈이 operation을 진행 중이라면 TRUE를 그렇지 않다면 FALSE를 반환한다.

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.55 RfidGetR1000MacReg

R1000의 레지스터 값을 읽는다.

```
BOOL RfidGetR1000MacReg (
    unsigned short Reg,
    unsigned int * pValue
);
```

Parameters

Reg

레지스터 주소(별도의 문서 참조)

pValue

레지스터 값이 저장될 변수

Return Values

성공적으로 수행되면 true를 리턴한다.

Notes

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.56 RfidSetR1000MacReg

R1000의 레지스터에 값을 기록한다.

```
BOOL RfidSetR1000MacReg (
    unsigned short Reg,
    unsigned int Value
);
```

Parameters

Reg

레지스터 주소(별도의 문서 참조)

Value

레지스터에 저장할 값이 저장된 변수

Return Values

성공적으로 수행되면 true를 리턴한다.

Notes

모듈 타입이 MODULE_TYPE_H1000일 때만 지원된다.

2.6.57 RfidSetTagFocus

Impinj Extension(Tag Focus) 기능을 Enable/Disable 한다.

EnableTagFocus 속성이 True로 설정되어 있는 동안은 InventoryTarget, Session 속성을 변경할 수 없다.

```
RFID_RESULT RfidSetTagFocus (
    BOOL bEnable,
);
```

Parameters

bEnable

Tag Focus 기능의 Enable/Disable 상태

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴한다.

Notes

모듈 타입이 MODULE_TYPE_I2000일 때만 지원된다.

2.6.58 RfidGetTagFocusStatus

Impinj Extension(Tag Focus) 기능의 Enable/Disable 상태를 읽는다.

EnableTagFocus 속성이 True로 설정되어 있는 동안은 InventoryTarget, Session 속성을 변경할 수 없다.

```
RFID_RESULT RfidGetTagFocusStatus (
    BOOL * pbEnable,
);
```

Parameters

pbEnable

Tag Focus 기능의 Enable/Disable 상태가 저장될 변수

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴한다.

Notes

모듈 타입이 MODULE_TYPE_I2000일 때만 지원된다.

2.6.59 RfidSetAirDura

모듈의 Tx Cycle 동작 시간을 조정한다.

```
RFID_RESULT RfidSetAirDura (
    RFIDAIRDURAPARAMS * pRfidParams
);
```

Parameters

pRfidParams

Tx Cycle의 on시간과 Off시간이 저장되어 있는 구조체의 포인터

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴한다.

Notes

MODULE_TYPE_I2000일 경우, Tx Cycle 동작시간의 기본값은 On(200ms), Off(0ms)이다.

2.6.60 RfidGetEuroMode

모듈 타입이 MODULE_TYPE_H1000이고 Europe항일 경우에 모듈의 동작 방식을 읽어 온다.

```
BOOL RfidGetEuroMode (
    unsigned int * pmode
);
```

Parameters

mode

1 : 4개 채널 hopping

2 : 4개 채널 중 1개 채널이 랜덤으로 선택되어 고정.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴 한다.

Notes.

2.6.61 RfidSetEuroMode

모듈 타입이 MODULE_TYPE_H1000이고 Europe항일 경우에 모듈의 동작 방식을 설정한다.

```
BOOL RfidSetEuroMode (
    unsigned int mode
);
```

Parameters

mode

1 : 4개 채널 hopping

2 : 4개 채널 중 1개 채널이 랜덤으로 선택되어 고정.

Return Values

성공적으로 수행되면 RFID_RESULT_SUCCESS를 리턴 한다.

Notes.